PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H03M 7/30

A1

(11) International Publication Number: WO 99/01940

(43) International Publication Date: 14 January 1999 (14.01.99)

(21) International Application Number: PCT/GB98/01937

(22) International Filing Date: 1 July 1998 (01.07.98)

(30) Priority Data:
9713921.6 1 July 1997 (01.07.97) GB

(71) Applicant (*for all designated States except US*): HEXAGEN TECHNOLOGY LIMITED [GB/GB]; 214 Cambridge Science Park, Milton Road, Cambridge CB4 4WA (GB).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): GILCHRIST, Michael, James [GB/GB]; 15 Willis Road, Cambridge CB1 2AQ (GB).

(74) Agent: HALLYBONE, Huw, George; Carpmaels & Ransford, 43 Bloomsbury Square, London WC1A 2RA (GB).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: BIOLOGICAL DATA

(57) Abstract

Using a whole byte to represent a monomer in a biological sequence is not the most efficient means of permanent storage. The invention relates to the compression of biological sequence data for electronic storage by utilising a sub–byte datatype for the storage or manipulation of biological sequence data in a programming language or a database. For nucleotide sequences, for example, 2 bits can be used to represent each monomer.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

BIOLOGICAL DATA

This invention relates to the compression of biological sequence data for electronic storage.

The nature of biological sequence data (*eg.* DNA and protein sequences) means that electronic storage is perfectly suited. Not only does the sheer volume of data necessitate large-scale storage, but electronic storage allows rapid and efficient searching of the data *eg.* for homologous sequences. Since the advent of initiatives such as whole genome sequencing, the amount of storage required has increased significantly. The storage requirement for the yeast genome, for instance, is huge. Whilst large capacity storage systems continue to fall in price, one of the rate-limiting steps when dealing with sequence data is the transfer from storage medium into memory (*eg.* hard drive into RAM) and any developments which significantly reduce the size of sequence data files would be welcomed.

Biological sequence data is typically represented in an alphabetic manner, rather than by chemical formula, with each letter representing a monomer unit in a biological polymer (Table I). For instance, DNA sequences are represented as strings of letters chosen from a simple four-letter "alphabet". Each A, C, G or T represents a monomer unit (nucleotide) in a DNA polymer. Similarly, proteins are made up of twenty different monomer units (amino acids), which have each been assigned single letter codes.

Because of its alphabetic nature, biological sequence data is naturally suited to electronic storage in alphabetic text form. Alphabetic text-based computer information is generally stored and manipulated using the *char* datatype, using 8 bits (1 byte) and a conventional file of biological sequence data is made up of a string of characters of datatype *char*. A conventional file of sequence data uses a single byte to represent each monomer, so the amino acid sequence of the glycogen synthase protein, for example, requires 737 bytes of storage using the one-letter amino acid code, and the corresponding DNA sequence requires 2211 bytes.

The *char* datatype, however, was designed for representing a full character set, including upper and lower case letters plus numbers, punctuation, and other characters, and each 8-bit *char* can represent 256 (2^8) different values. Using the *char* datatype and alphanumeric characters to store DNA sequences therefore fails to utilise 252 of the available values. Similarly, protein sequences waste 236 values. Other datatypes which are in common usage for data storage include *int* (16 bits), *long* (32 bits), *float* (32 bits), although this may vary from machine to machine.

The DNA and RNA alphabets each consist of 4 letters and, rather than storing these sequences in alphanumeric form using strings of *char* datatype, using a sub-byte datatype would enable a significant storage saving. Degenerate nucleic acid sequence information (which can be represented using a 16 letter alphabet) and protein sequences could also be treated in this way. It would therefore be useful to define a sub-byte datatype in order to take advantage of the small size of the biological alphabet.

The commonly used *blast* sequence comparison program converts single byte *char* data into a half-byte working space whilst manipulating data. This is a temporary measure, however, and data is not stored in this manner using a specific sub-byte datatype.

The invention is based upon the realisation that using a whole byte to represent a monomer in a biological sequence is not the most efficient means of permanent storage.

According to the invention, there is defined a sub-byte datatype for the storage or manipulation of biological sequence data in a programming language or a database.

The invention also provides a programming language or a database which utilises a sub-byte datatype for the storage or manipulation of biological sequence data.

According to a further aspect of the invention, there is provided the use of a sub-byte datatype in the storage or manipulation of biological sequence data.

By "sub-byte" it is meant fewer than 8 bits.

The datatype may be intrinsic to a program or programming language, or it may be user-defined. The invention is not limited, however, to situations where a formal datatype must be defined.

According to a further aspect of the invention, there is provided a computer program which stores biological sequence data using fewer than 8 bits to represent each monomer in said sequence data.

The invention also provides a file containing biological sequence data, wherein each monomer in said sequence data is represented using fewer than 8 bits.

According to a further aspect of the invention, there is provided a method for compressing biological sequence data, comprising representing each monomer in said sequence data by using fewer than 8 bits.

The invention also provides a method for reducing the size of a file in which biological sequence data is represented using 8 or more bits per monomer, comprising replacing the representation of each monomer with a representation using fewer than 8 bits.

According to a further aspect of the invention, there is provided a computer programmed to store biological sequence data by using fewer than 8 bits to represent each monomer in said sequence data.

According to a further aspect of the invention, there is provided a computer comprising means for alphabetic entry of biological sequence data, means to convert said sequence data into a format wherein each monomer unit is represented using fewer than 8 bits and, preferably, means to store said data.

According to a further aspect of the invention, there is provided a storage medium holding biological sequence data, wherein said sequence data is stored using fewer than 8 bits to represent each monomer in said sequence data.

The storage medium may be in any appropriate form, such as a floppy disk, a CD-ROM, or a fixed disk drive.

According to a further aspect of the invention, there is provided a method for transmitting biological sequence data, comprising compressing the data by representing each monomer in said sequence data by using fewer than 8 bits before transmission, for instance over a network.

According to a further aspect of the invention, there is provided biological sequence data which has been electronically stored using less than 8 bits to represent each monomer in said sequence data.

The biological sequence data may be of any suitable kind, such as DNA sequence, RNA sequence, and protein or polypeptide sequence.

It will be apparent that nucleic acid sequences can be represented using 2 bits to represent each monomer (nucleotides A, C, G, or T/U). Accordingly, a 2 bit datatype may be defined according to the invention for the storage or manipulation of nucleic acid sequences. Such a datatype is referred to herein as *base*.

By representing each nucleotide in a nucleic acid sequence by using only 2 bits, 4 nucleotides can be stored in a single byte. This represents a 75% compression compared with the conventional representation of each nucleotide using a single byte.

Where a nucleic acid sequence is not definite, more than 2 bits are required to represent each nucleotide. For instance, where a nucleotide has not been unequivocally determined, the symbol "N" is used according to IUPAC convention. The alphabet of this IUPAC convention (Table I) has 16 members. This can be conveniently represented using 4 bits per member. Accordingly, a 4 bit datatype may be defined according to the invention for the storage or manipulation of degenerate or uncertain nucleic acid sequences. Such a datatype is referred to herein as *longbase*.

By representing each nucleotide in a sequence by using 4 bits, 2 nucleotides can be stored in a single byte. This represents a 50% compression compared with the conventional representation of each nucleotide using a single byte.

As an alternative to using 4 bits to represent degenerate or uncertain nucleic acid sequences, under certain circumstances these features may be accommodated where 2 bits are used, as in *base*. For instance, where a DNA sequence is stored in a data file using 2 bits per nucleotide, parallel files could be utilised which contain "modifying" data to qualify details in the sequence file. For instance, the second file may contain an indication that whilst nucleotide 221 is given as guanine in the sequence file, in fact it may be any purine. Obviously, the choice of using such a "modifying" file or using more than 2 bits to represent the sequence depends on the particular situation, but the choice is routine.

It will further be apparent that protein sequences require at least 5 bits to represent each monomer (20 amino acids) since 2^4=16 and 2^5=32. Whilst this is encompassed within the invention, 5 bits is an awkward length, being an odd number. 6 bits is more convenient and, furthermore, this allows a degree of degeneracy to be incorporated into the sequence (2^6=64). Accordingly, a 6 bit datatype may be defined according to the invention for the storage or manipulation of protein sequences. Such a datatype is referred to herein as *aminoacid.*

By representing each amino acid in a protein sequence by using 6 bits, 4 amino acids can be stored in 3 bytes. This represents a 25% compression compared with the conventional representation of each amino acid using a single byte.

The degree of degeneracy incorporated into a 6-bit representation or datatype also allows an amino acid to be represented in terms of codons, of which there are 64. A datatype used in this way is referred to herein as *codon*. Each single *codon* value represents a single codon, which inherently also defines an amino acid. In effect, the *codon* datatype represents three *base* entries, just as a codon is made up of three nucleotides. By using 6 bits to represent each codon, 4 codons can be represented in 3 bytes. This represents a 75% compression compared with the conventional representation of each codon using 3 bytes. It will also be appreciated that a full byte could be used to represent each codon, which would allow a degree of degeneracy and would represent a 67% compression compared with using 3 bytes to represent each codon.

It should be borne in mind that the various datatypes and compressions described above may not be suitable in all circumstances. For example, the programming language C requires a string to have a NULL terminator. This is not possible with the *base* datatype, for instance, because all of the 4 possible values (permutations of 2 bits) are used to represent information, which does not allow a terminator to be represented.

Similar caveats apply to *longbase*. The IUPAC convention uses 15 representations for a DNA or RNA sequence, which does allow the sixteenth permutation to represent a terminator. In certain circumstances, however, a value may be needed to represent a gap (representing an unknown sequence of unknown length) which would remove the possibility of having a terminator. The *codon* datatype is also "full" since each of the 64 available values represents a codon.

Whilst these datatypes may not be universally applicable, however, they are not without utility since not all programming languages or databases have such a terminator requirement. A further problem in using the datatypes of the invention in languages such as C is the international ANSI standard which does not recognise these datatypes. However, new languages, such as Java which is still in early development, currently have less strict standards and may be amenable to the introduction of new datatypes at this stage.

TABLE I

IUB/IUPAC standard biological sequence codes

Single letter nucleotide codes

A	Adenine	C	Cytosine
G	Guanine	T	Thymine
U	Uracil		

Degenerate nucleotide codes

In addition to the five above codes:

N	any (A/C/G/T)	R	puRine (G/A)	Y	pYrimidine (T/C)
K	Keto (G/T)	M	aMino (A/C)	S	Strong (G/C)
W	Weak (A/T)	B	not A (C/G/T)	D	not C (A/G/T)
H	not G (A/C/T)	V	not T (A/C/G)		

Single letter amino acid codes

A	Alanine	C	Cysteine	D	Aspartate
E	Glutamate	F	Phenylalanine	G	Glycine
H	Histidine	I	Isoleucine	K	Lysine
L	Leucine	M	Methionine	N	Asparagine
P	Proline	Q	Glutamine	R	Arginine
S	Serine	T	Threonine	V	Valine
W	Tryptophan	Y	Tyrosine		

In addition:

B represents asparagine or aspartate *ie.* N or D

Z represents glutamine or glutamate *ie.* Q or E

U represents selenocysteine

X represents "any amino acid" or "unknown"

* represents a translation stop

– represents a gap of indeterminate length

CLAIMS

1. A sub-byte datatype for the storage or manipulation of biological sequence data in a programming language or a database.

2. A programming language or a database which utilises a sub-byte datatype for the storage or manipulation of biological sequence data.

3. The use of a sub-byte datatype in the storage or manipulation of biological sequence data.

4. A file containing biological sequence data, wherein each monomer in said sequence data is represented using fewer than 8 bits.

5. A method for compressing biological sequence data, comprising representing each monomer in said sequence data by using fewer than 8 bits.

6. A method for reducing the size of a file in which biological sequence data is represented using 8 or more bits per monomer, comprising replacing the representation of each monomer with a representation using fewer than 8 bits.

7. A computer programmed to store biological sequence data by using fewer than 8 bits to represent each monomer in said sequence data.

8. A storage medium holding biological sequence data, wherein said sequence data is stored using fewer than 8 bits to represent each monomer in said sequence data.

9. Biological sequence data which has been electronically stored using less than 8 bits to represent each monomer in said sequence data.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 98/01937

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H03M7/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H03M

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	WO 97 31327 A (MOTOROLA INC ;REBER WILLIAM L (US); PERTTUNEN CARY D (US)) 28 August 1997 see page 5, line 21 - line 24 ---	1-9
A	US 4 701 744 A (DEVILBISS WARREN C) 20 October 1987 see the whole document -----	1

□ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of theinternational search 29 September 1998	Date of mailing of the international search report 08/10/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Feuer, F

1

INTERNATIONAL SEARCH REPORT

In. itional Application No
PCT/GB 98/01937

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9731327 A	28-08-1997	AU 1414197 A	10-09-1997
US 4701744 A	20-10-1987	AU 602512 B	18-10-1990
		AU 7046387 A	01-10-1987
		CA 1283483 A	23-04-1991
		DE 3787985 D	09-12-1993
		DE 3787985 T	19-05-1994
		EP 0240242 A	07-10-1987
		FI 871243 A,B,	28-09-1987
		JP 62237496 A	17-10-1987
		KR 9513258 B	26-10-1995